

Sheet Metal Laser Cutting Tool Path Generation: Dealing with Overlooked Problem Aspects

Reginald Dewil^{1,a}, Pieter Vansteenwegen^{1,b} and Dirk Cattrysse^{1,c}

¹KU Leuven, Centre for Industrial Management/Traffic & Infrastructure, Celestijnenlaan 300a, 3001 Leuven, Belgium

^areginald.dewil@kuleuven.be, ^bpieter.vansteenwegen@cib.kuleuven.be,

^cdirk.cattrysse@cib.kuleuven.be

Keywords: laser, cutting, tool path, Computer aided manufacturing (CAM).

Abstract. This paper deals with non-trivial problem aspects of laser cutting tool path generation that, to the best of our knowledge, received relatively little attention in the scientific literature. It is shown that some aspects such as plate edge nesting, skeleton and remnant cutting, and clamp positioning can be modeled and solved with little additional effort using existing tool path algorithms. However, concepts such as collision avoidance, pre-cut optimization, and bridge utilization prove to be more challenging and will require more profound algorithmic adjustments if these have to be taken into account fully. An even harder problem aspect is generating tool paths that are thermally feasible. Since laser cutting introduces net heat into the metal sheet, the metal sheet tends to heat up as the cutting progresses. Quality deterioration can occur if the laser spends too much time cutting in the same region.

It is shown how to model the easy problem extensions in order to handle them using existing problem approaches and solution approaches are suggested to tackle the harder concepts. In addition, a proof of concept is presented that shows that thermal feasible tool paths can be generated through a multi-start heuristic utilizing a thermal penalty function. A finite difference method iteratively (or concurrently dependent on the used heuristic) evaluates the thermal feasibility and updates the penalty function.

Introduction

Laser cutting has become one of the major technologies used to separate unfolded parts from metal sheets and plates. A part consists of an outer contour and possibly a set of inner contours (holes). A contour is, itself, composed of a set of elements (lines or arcs). The tool path is subject to a set of precedence constraints originating from the fact that once a contour is fully cut, it detaches from the rest of the plate, can possibly shift its position, and thus making additional cuts within this area inaccurate. Therefore, a precedence constraint states that when a contour is fully cut (when its final element is cut), no uncut elements may remain within the contour. Or put otherwise: *all elements of an inner contour need to be cut before the last element of its outer contour is cut*. Another set of constraints originates from common cuts. Common cuts are the result of nesting parts so close to one another that they share an element. This reduces waste material, but also requires only one cutting operation instead of two, greatly reducing the total cutting time. However, common cuts also introduce the additional constraint that *no common cut is allowed to simultaneously close both its contours* [1].

Traditionally, tool path algorithms have considered the so-called Continuous Cutting path Problem (CCP) or the Generalized Traveling Salesman Problem (GTSP). In both the CCP and the GTSP, once the laser starts to cut a contour, it needs to cut the contour completely before moving on to another contour. In the CCP, the laser head can initiate a contour at any location, while, in the GTSP, a set of possible entry/exit locations is defined beforehand. A recent literature overview can be found in Dewil [2].

Dewil et al. [1][3], show that significant gains can be achieved by considering the tool path problem as an End point Cutting Problem (ECP). Here, the laser head is not constrained to finish a contour in one pass. It is allowed to stop cutting a certain contour and to go cut other contours

before moving back and finishing the original contour. This process is referred to as *pre-empting* a contour. In the ECP, accommodating precedence constraints originating from inner-outer contour relations and common cuts in addition to incorporating air move, piercing, pre-cut, and sharp angle macro costs requires problem-specific approaches to efficiently generate good and feasible tool paths.

To the best of the authors' knowledge, generating tool paths incorporating parts nested on the plate edge, clamp positions, skeleton cuts, collision avoidance, and pre-cut optimization has not been investigated in an operations research context. Bridges have been instigated in the context of pierce point minimization by Manber and Israni [4], but only for contour groups connected through common cuts. Only three papers explicitly consider thermal effects when generating tool paths. Both Han and Na [5][6] and Kim et al. [7] consider the GTSP version of the tool path problem. In both solution approaches the thermal effects are over-simplified and both approaches do not guarantee thermal feasible tool paths. The remainder of this paper is structured as follows. In the next section, additional problem aspects are discussed which by some clever preprocessing can be handled by existing algorithms. The third section deals with more complicated problem extensions such as collision avoidance, pre-cut optimization, and bridge utilization and the fourth section presents an approach to generate thermal feasible tool paths. The last section formulates our conclusions and offers further research possibilities.

Removing Complexity

It will be shown in this section that several problem aspects can be remodeled using the abovementioned two types of precedence constraints and therefore do not introduce additional complexity for optimizing the cutting tool path when modeling it in this way.

First of all, contours can be nested on the plate edge. As shown in figure 1, this means that one or more elements of a contour are located on the plate edge. As a consequence, these elements do not have to be cut which leads to great time savings. Furthermore, adjacent elements can be initiated from the plate edge at a highly reduced piercing cost. The plate edge elements can be considered as elements that are already cut and consequently, inner-outer contour relations can be handled in the same way for these contours as for regular contours. However, such plate edge nesting can give rise to enclosed areas. If parts would be nested in these areas, care should be taken that this enclosed area does not detach from the rest of the plate before all elements within this area have been cut. Similarly, even some elements of the contour nested on the plate edge will have to be cut before the contour is detached from the rest of the plate. For example, in figure 1(a), all elements within the encircled enclosed area, but also the arc element of the contour itself will have to be cut before the last element of all bolded elements is cut. Or put otherwise, the arc element cannot be the last element to be cut from the contour. The same is true in figure 1(b), where several areas on the plate edge are enclosed. None of the elements of all these areas are allowed to close their contours.

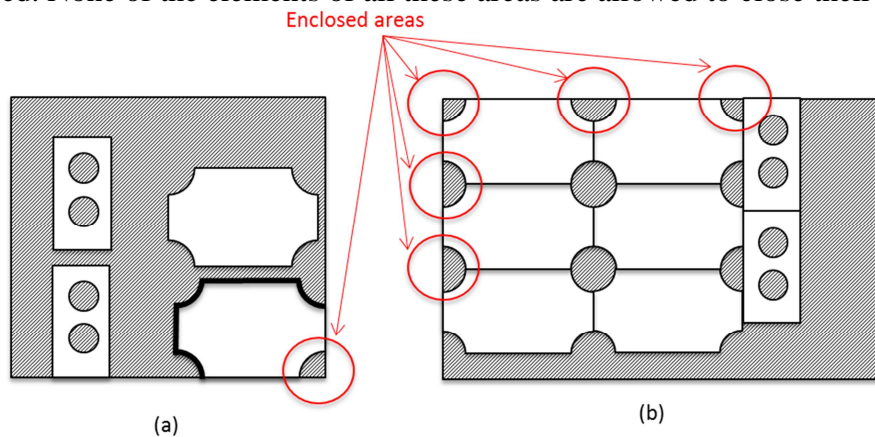


Figure 1: Parts can be nested on the plate edge[2].

This can readily be modeled by defining a contour for each of these areas and declaring these arc elements as common cuts. Then, they will be subject to the abovementioned common cut precedence constraints which ensure that a common cut is not allowed to simultaneously close both its contours.

When all elements on a plate or sheet are cut, a skeleton of waste material is left. In order to facilitate the disposal of this skeleton, it is often cut into smaller pieces. Small time savings can be achieved including these *skeleton cuts* in the regular tool path; i.e. while cutting the nested parts. Similarly, the *remnant cut* can also be included in the regular cutting process. A metal sheet is often not completely nested. When this occurs, the nesting is executed in such a way as to maximize a reusable rectangular area called the *remnant area*. In order to separate this area from the skeleton, one or two cuts need to be executed and it are these cuts that can be incorporated in the regular cutting process as well. As can be seen in figure 2, both the skeleton and remnant cuts can be seen as common cuts. Additionally, these common cuts are part of (newly defined) contours that might be nested on the plate edge. These contours, in turn, are subject to additional inner-outer contour relations with the contours already nested on the metal plate. A (computationally inexpensive) preprocessing phase can place these skeleton cuts, identify the newly formed contours and define the additional precedence constraints.

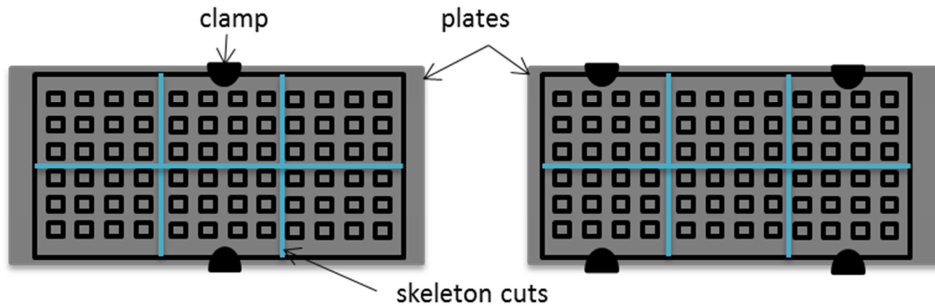


Figure 2: Skeleton cuts can be considered as regular (common) cuts[2]

However, skeleton and remnant cuts introduce additional precedence constraints. Suppose for example that a plate has a single skeleton cut dividing the plate into two regions right in the middle. If one cuts this skeleton cut, it is unclear which region detaches from “the rest of the plate”. Elements within the detached area are not able to be cut since this area might have shifted. Therefore, one or more regions on the plate will have to be declared as *fixed* i.e. these regions can never shift position and thus any elements within will always be accessible to be cut.

Fixed regions can be declared on the basis of the clamp positions as shown in figure 2¹. Common cut constraints don’t apply to elements that are part of a contour that is held in place. For example, in figure 2(a), common cut constraints only apply on the outermost two horizontal common cuts. The other common cuts are unconstrained. In figure 2(b), however, all common cuts are unconstrained except for the middle horizontal common cut.

In addition, contours nested in fixed regions also don’t have to be cut before its outer contour (the one that is composed out of the skeleton cuts and, possibly, the plate edge). A (computationally inexpensive) preprocessing phase can easily identify the clamp positions and assign the *fixed* status to the relevant contours and consequently remove the inner-outer contour and common cut precedence constraints.

Non-trivial extensions

Collision avoidance. The first non-trivial extension involves avoiding collisions between the laser head and the work pieces. When small parts are cut and they detach from the rest of the plate, it can occur that the parts tip and jut out of the surface of the sheet, as shown in the side view of a plate in figure 3. If the laser head then moves over this area, it can collide with this part, which can

¹ In practice, fixed regions can also be declared if the region is sufficiently large and a supporting grid is present.

result in damage to the laser head and/or to the part. A possible addition to the tool path problem then involves the trade-off between taking a longer route avoiding a possible collision area and executing a lift/lower move. A lift/lower move entails lifting the laser head before moving over the area and lowering the laser head afterwards. Both moves require additional time to execute. However, the inclusion of this decision-making in the tool path algorithm entails a very large increase in computational complexity as suddenly the underlying graph becomes dynamic i.e. the travel time between two nodes changes as the path is being constructed.

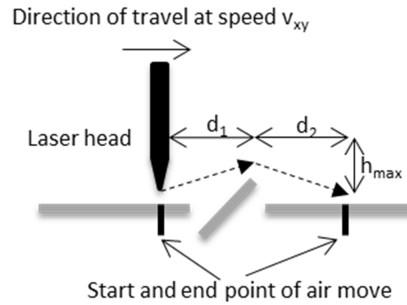


Figure 3: Some lift/lower commands will incur additional costs. Only in the cases where the possibly tipped part is close to the start and/or end location of an air move, delay costs will be incurred[2].

However, if a supporting grid is present on the laser cutting machine, only contours that are smaller than twice the grid spacing pose the risk of tipping over since if they are larger, they will be supported. Additionally, the additional time required to lift or lower the laser head is relatively low. Therefore, the added complexity of including these calculations in the tool path algorithms doesn't outweigh the possible gains, and thus, a post-optimization procedure suffices to avoid collisions.

Such a procedure involves going over the proposed tool path and for each air move, the procedure would check whether this movement crosses a contour that is already cut, and that is smaller than twice the grid size. If this would be the case, additional checks would have to occur on how to issue the lift/lower command. In figure 3, the laser head needs to move over a tipped contour. In order to do this, it will need to reach a height of h_{max} . If the time required to travel the d_1 distance is lower than the time required to travel the h_{max} distance, a regular air movement can be executed to the (x,y,z) point corresponding to the (d_1, h_{max}) point. Else, an air movement at reduced acceleration, deceleration, and maximum speed in the x,y -plane needs to be executed while the laser head is lifted. A similar analysis has to occur on the downward part of the air move.

However small the probability of collision might be (especially if the tool path is determined using a pre-emption strategy as long air moves are usually not present in such tool paths), this procedure completely removes the risk of collisions with at most a slight increase in air move execution time.

Pre-cut optimization. Pre-cutting is the process of partly (a couple of millimeters) cutting an element that will be cut later on. This can be used to avoid a time-consuming piercing or is sometimes required by a special precedence constraint when contours are connected through common cuts. A clear description of this precedence constraint can be found in Dewil et al. [1]. If multiple pre-cuts have to be made at the same junction, the pre-cuts can be optimized in order to minimize total sharp angle costs. Consider figure 4, where the laser is cutting elements A and D, and needs to place pre-cuts at elements B and C. If it first pre-cuts element B and then element C, three sharp angle macros will be incurred (4b). However, if first C is pre-cut and then B, only one sharp angle macro will be incurred (4c).

To the best knowledge of the authors, no tool path algorithm has been proposed in the literature that includes this pre-cut sub optimization in the tool path algorithm. However, it is trivial to include in a post-optimization procedure. Given n pre-cuts at an intersection, there are $n!$ orderings possible. However, in practice, no single intersection will require more than 5 pre-cuts. Therefore, a simple brute-force approach will suffice to find the optimal ordering per intersection.

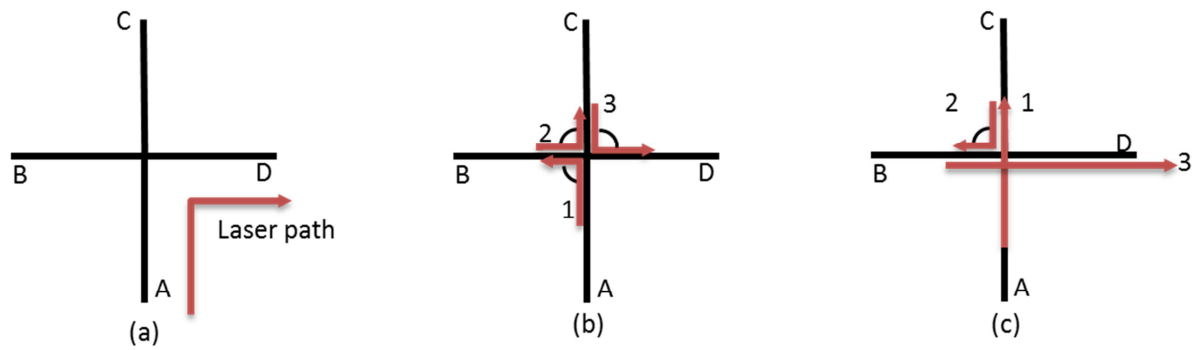


Figure 4: If multiple pre-cuts need to be made at the same junction, the order of making the pre-cuts can be optimized[2].

Bridges. In order to avoid expensive piercings, so-called bridges can be used. When moving between two contours across a waste area, the laser can continue cutting instead of the combined process of stopping the cutting operation, executing an air move, piercing the plate, and cutting a lead-in in order to reach the second contour. Although piercings are very expensive, air moves are considerably cheaper than cut moves. It follows that a bridge cannot be longer than a certain length before a combination of piercing and air move becomes cheaper, dependent on the material, plate thickness, and machine parameters. Adding this extension to the problem significantly complicates the problem as this would introduce elements that may or may not be cut. If these elements are cut, they might create new contours, which themselves can imply new common cut and inner-outer contour precedence constraints as depicted in figure 5. Incorporating these dynamic elements into existing heuristics is not trivial. However, dealing with bridges in either a preprocessing or post-optimization procedure is more tractable.

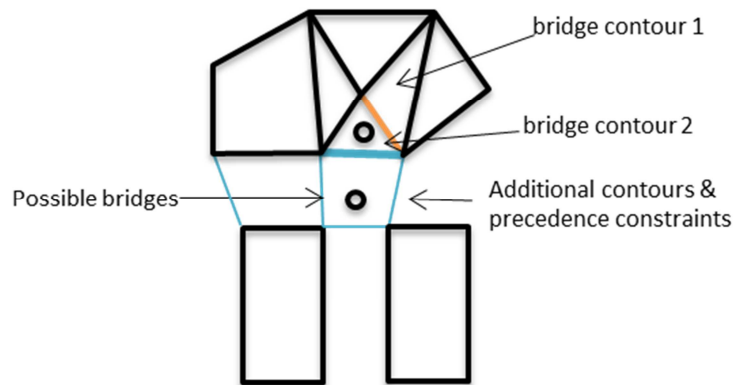


Figure 5: Introducing bridges, introduces additional contours[2].

A preprocessing phase can identify relevant bridges beforehand and define them as normal cutting elements. If the correct additional contours are identified and the relevant elements are designated as common cuts, a tool path can be determined using the proposed heuristics. Initially, an operator (or separate procedure) needs to identify which air move and piercing combinations (of all possible moves) can profitably and feasibly be replaced by a bridge. Deciding where to place the bridges can be done using the algorithm of Manber and Israni [4]. The algorithm of Manber and Israni selects those bridges that minimize the total number of remaining piercings. A drawback of this method is that it does not take the total bridge length into account i.e. selecting two long bridges is equivalent to selecting two short bridges in order to eliminate two piercings. But the method can easily be adapted to heuristically consider bridge lengths while minimizing the number of piercings.

A second approach lies in determining the tool path while ignoring bridges. Then, in a post-optimization procedure, all air move and piercing combinations that can profitably be replaced by a bridge, without violating precedence or thermal feasibility constraints, are identified and replaced. This post optimization will result in more bridge introductions with a tool path that has been constructed on the basis of a pre-empt strategy than one that has been constructed on the basis of a

GTSP strategy since in a pre-empt based tool path more piercing and short air move combinations are present.

Generating thermal feasible tool paths

Since the laser cutting process introduces net heat into the metal sheet, the metal sheet tends to heat up as the cutting process progresses. Excessive heating, however, can cause quality deterioration. Figure 6 shows the results of consecutive cutting tests performed, phased in time and space, by Duflou et al. [8]. First, line 1 is cut. Then, line 2 is cut, and lastly, line 3 is cut. It can be seen that the length across which a stable cut can be realized becomes systematically shorter. This preheating effect is even more pronounced in sharp angles where a corner burn-off can occur.

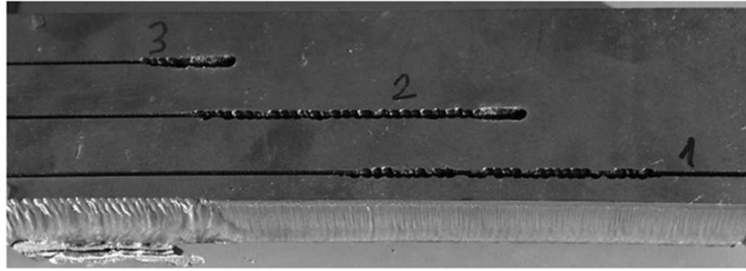


Figure 6: Excessive plate pre-heating causes quality deterioration [8].

This implies that the tool path will need to take these thermal effects into account i.e. avoid cutting in an overheated area. This means that if preceding cut moves have brought an area up to a threshold temperature, the laser head will need to cut in another area, allowing the overheated area to cool down.

Han and Na [5][6] used a finite difference method (FDM) to determine penalties that can be added to the travel cost when moving from one contour to another contour. As a consequence, Han and Na end up with a regular GTSP and solve this using Simulated Annealing. The main criticisms on their approach are:

1. The temperature at a piercing location is assumed to be only affected by the contour that is cut just before it. Furthermore, it is only the nearest element that is responsible for the heat flows. Certainly, all cutting affects the temperatures across the entire plate.
2. Convective and radiative heat flows are not considered in the FDM.
3. There is no actual feasibility checking. The penalties are determined beforehand and do not change during optimization.

Kim et al. [7], on the other hand, do execute a complete FDM temperature analysis every time a feasible path is determined through a Genetic Algorithm. If a part cannot be cut feasibly, it is put on hold and cut later. The main criticisms on their approach are:

1. Only the temperature at the piercing location during the piercing execution, is used to determine if a part can be cut feasibly. Certainly, a very large part can have a feasible piercing temperature at one side of the part, while the other side is at a critical temperature making cuts infeasible.
2. The partial path up to the first infeasibility is never altered. If this partial path has enclosed thermal energy in an area where a contour still needs to be cut and the time required to cool this area down below the threshold temperature is larger than the time required to cut all other parts, it will not be able to find a feasible path. The only way to find a feasible path then is to introduce waiting times.
3. Convective and radiative heat flows are not considered, thus making waiting times also less effective.
4. Cut kerfs are not considered adiabatic boundaries.

Addressing the main criticisms of both methods, an ideal method for the plate cutting path problem would entail that the thermal effect of every cutting operation (piercing, cutting, pre-cutting, sharp angle macros and piercings) is evaluated across the entire plate. Convective and

radiative heat flows are considered and the adiabatic boundary of the cut kerf is taken into account. If an infeasible path is detected, the thermal violation itself should guide the search in following iterations.

This can be accomplished by using elements from the solution approaches of both Han & Na and Kim et al.. A construction heuristic, such as the PFr heuristic of Dewil et al. [1], can be augmented with *thermal penalties* similar to the ones used by Han & Na. But instead of thermal penalties between entry/exit points on contours, these would be defined between individual cuts (combination of element and cutting direction). After running this *thermal augmented construction heuristic* (C^T), a temperature evaluation can be executed similar to the one executed by Kim et al. The temperatures should be evaluated on the cut level, take the relevant convective and radiative heat flows into account, and consider the adiabatic boundaries of cut kerfs. If the temperature during the execution of a certain operation in the path exceeds the critical temperature (T_{crit}) for that specific feature (piercing, sharp corner, ..), material, and thickness combination, the *thermal penalties* up to that point in the partial path should be increased as these have led to the infeasibility. Consequently, in the next iteration, this partial path is then less likely to be chosen. This can be seen as similar to the pheromone updating rule of an ant colony optimization (ACO) algorithm [9]. But instead of using the objective function value to update the pheromones, the relative temperature violation ($\frac{T}{T_{crit}}$) can be used to update the *thermal penalties*.

An FDM divides the metal sheet in small cells and, in each small time-step, both the heat flow between cells is determined, as well as the temperature change of each cell resulting from those heat flows. Taking conductive, convective, and radiative heat flows into account, the temperature change of a cell (ij) between successive iterations (g and $g+1$) can be determined by equation (1).

$$T_{ij}^{g+1} = T_{ij}^g + \Delta t \left[\alpha \frac{\sum_{adj} T_{ijadj}^g - 4T_{ij}^g}{a^2} - \frac{(h_{top} + h_{bot})(T_{ij}^g - T_{env}) + \epsilon \sigma (T_{ij}^{g4} - T_{env}^4)}{\rho d c_p} \right] \quad (1)$$

T_{ij}^{g+1} and T_{ij}^g equal the temperatures of cell (ij) in iteration g and $g+1$, Δt equals the time-step between iterations, α equals the thermal diffusivity constant, T_{ijadj}^g equals the temperature of an adjacent cell to cell ij in iteration g , a equals the cell width, $h_{top} + h_{bot}$ equal the convective heat transfer coefficient for the top and bottom side of the plate respectively, T_{env} equals the ambient temperature, ϵ equals the material's emissivity constant, σ equals the Stefan-Boltzman constant, ρ equals the material's density, d equals the plate thickness, and c_p equals the material's specific heat constant. Additional equations are required to model the net heat input during cutting and piercing which can be derived from the work of Schulz et al. [10].

As mentioned, if an infeasibility in the FDM evaluation procedure is detected, the penalties between cuts should be updated. Similar to the myriad ways of pheromone updating rules in ACO, multiple updating schemes for heat penalties can be constructed. Ideally, when a thermal violation is detected, the thermal penalties should be adjusted taking the degree of violation into account. Furthermore, the last decisions leading up to the violation have probably contributed more to the violation and, as such, should be penalized more (figure 7).

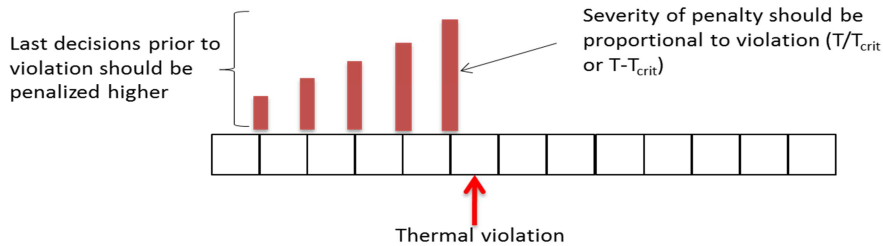


Figure 7: Proposed thermal penalization scheme: thermal penalties should be proportional to the thermal violation and inversely proportional to the index difference[2].

A straightforward rule for updating the thermal penalties φ can be of the following form: $\varphi'_{ij} = \varphi_{ij} + \frac{T}{T_{crit}} \xi^{(I_{viol}-I_j)}$, with φ_{ij} being the thermal penalty between cut i and cut j , ξ being a

parameter between 0 and 1 which controls the responsiveness of the updating scheme, I_{viol} being the path index of the thermal violation and I_j being the path index of cut j . A lower ξ results in a lower responsiveness to thermal violations. Because of the exponential nature of the penalty updating scheme, this is especially the case for decisions that are made many indices away prior to the violation.

The presented thermal penalty based multi-start heuristic was tested as a proof-of-concept on an instance containing 656 elements nested on a 2000mm x 1500 mm x 6 mm steel plate. A tool path determined by the PFr heuristic [1] requires 4266.5 seconds to execute all piercings, cut moves, air moves, lead-ins, lead-outs, sharp angle macros, and pre-cuts. The proof-of-concept tests showed that this approach can quickly lead to thermal feasible tool paths with a minimal increase in tool path execution time. However, the convergence times heavily depend on the ξ parameter as well as the chosen resolution of the FDM evaluation.

Summary

In this article, several extensions to the tool path problem were introduced and solutions approaches were proposed. Several extensions, if modeled properly, can be included in existing heuristics. For others such as collision avoidance, pre-cut optimization, and bridges, their inclusion would increase the computational complexity while its impact on the total tool path cost would be limited. Therefore, computationally inexpensive heuristic post-optimization procedures are proposed for each of these extensions. Thermal effects can be taken into account using a penalty-based multi-start heuristic. Between successive calls of the construction heuristic, an FDM can be used to evaluate the plate-wide temperatures during cutting. If a possible thermal defect is detected, the thermal penalties in the tool path up to the defect are increased so as to decrease the probability of the same tool path being chosen in the next iteration. Initial tests have shown that this approach leads to thermal feasible tool paths in a limited number of iterations. However, further research is required to test this approach on different construction heuristics, to speed up the FDM evaluation, and since the process is highly parameter-sensitive, to execute a structured parameter tuning.

References

- [1] R. Dewil, P. Vansteenwegen, and D. Cattrysse, "Construction heuristics for generating tool paths for laser cutters," *Int. J. Prod. Res.*, vol. 52, no. 20, pp. 5965–5984, 2014.
- [2] R. Dewil, "On generating tool paths for laser cutters," KU Leuven, 2014.
- [3] R. Dewil, P. Vansteenwegen, D. Cattrysse, T. Vossen, and M. Laguna, "An Improvement heuristic framework for the laser cutting tool path problem," *Int. J. Prod. Res.*, 2014.
- [4] U. Manber and S. Israni, "Pierce Point Minimization and Optimal Torch Path Determination in Flame Cutting," *J. Manuf. Syst.*, vol. 3, no. 1, pp. 81–89, 1984.
- [5] G. Han and S. Na, "A Study on Torch Path Planning in Laser Cutting Processes Part 1: Calculation of heat Flow in Contour Laser Beam Cutting," *J. Manuf. Process.*, vol. I, no. 1, pp. 54–61, 1999.
- [6] G. Han and S. Na, "A Study on Torch Path Planning in Laser Cutting Processes Part 2 : Cutting Path Optimization," *J. Manuf. Process.*, vol. 1, no. 1, pp. 62–70, 1999.
- [7] Y. Kim, K. Gotoh, and M. Toyosada, "Global cutting-path optimization considering the minimum heat effect with microgenetic algorithms," *J. Mar. Sci. Technol.*, vol. 9, no. 2, pp. 70–79, 2004.
- [8] J. R. Duflou, J. Kruth, J. De Keuster, K. Y. Sastry, and R. F. De Graaf, "On CO2 Laser Cutting of Thick Steel Plates," in *Proceedings of the 10th Int. Conf. on Sheet Metal*, 2003.
- [9] M. Dorigo, V. Maniezzo, A. Coloni, and A. Coloni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions Syst. Man, Cybern. Part-B*, vol. 26, no. 1, pp. 1–13, 1996.
- [10] W. Schulz, D. Becker, J. Franke, R. Kemmerling, and G. Herziger, "Heat conduction losses in laser cutting of metals," *J. Phys. D. Appl. Phys.*, vol. 26, pp. 1357–1357, 1993.